BERT 使用示例

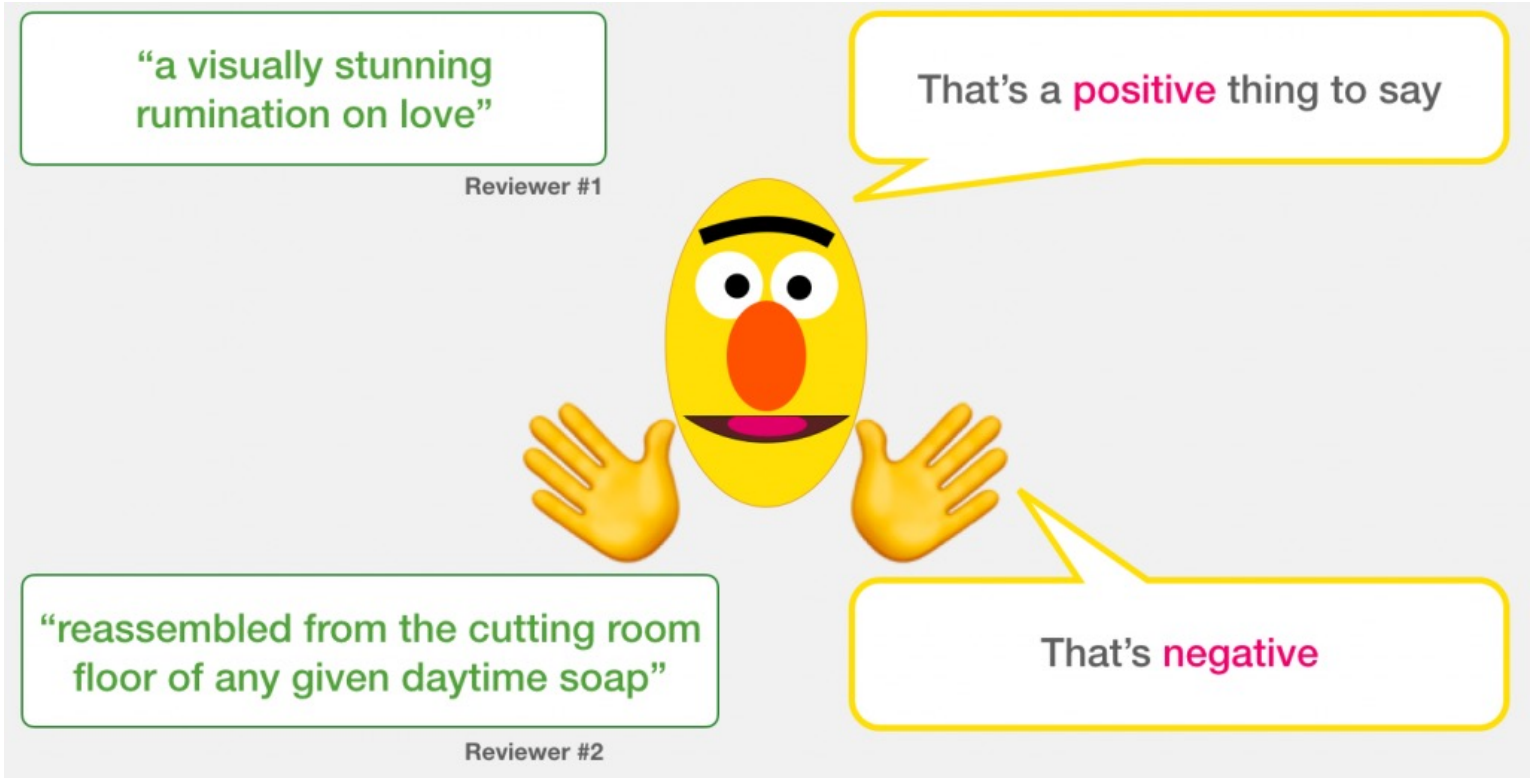
谢海华 助理研究员

北京雁栖湖应用数学研究院

大数据及人工智能

2022.04

文本情感分析


“a visually stunning rumination on love”
Reviewer #1
That’s a positive thing to say
“reassembled from the cutting room floor of any given daytime soap”
Reviewer #2
That’s negative

数据集：SST2

电影评论的数据集。用标签 0/1 代表情感正负：

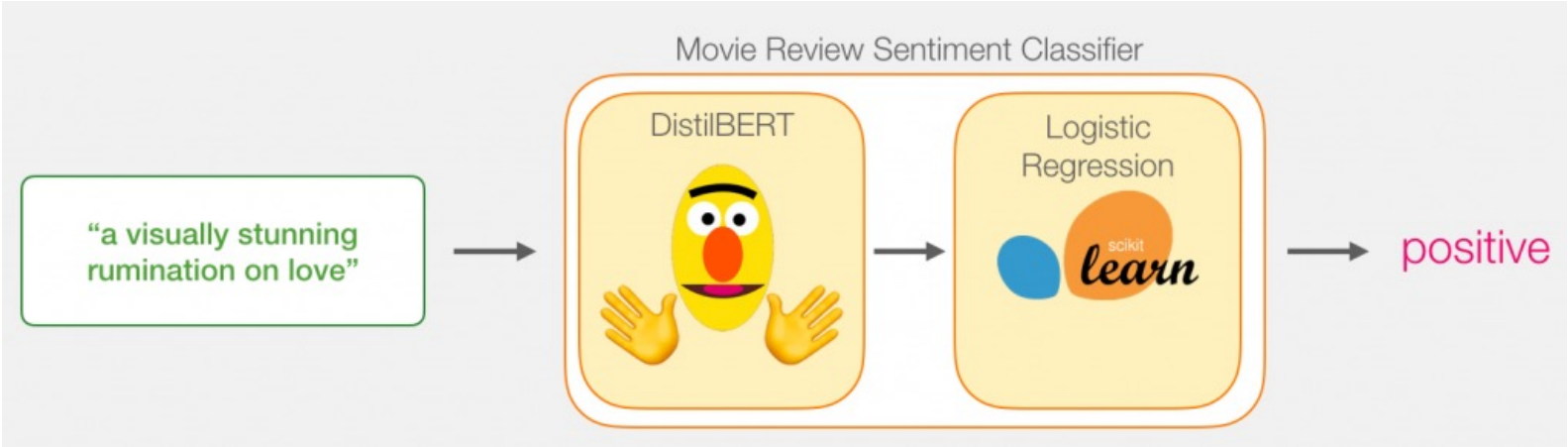
	0	1
0	a stirring , funny and finally transporting re...	1
1	apparently reassembled from the cutting room f...	0
2	they presume their audience wo n't sit still f...	0
3	this is a visually stunning rumination on love...	1
4	jonathan parker 's bartleby should have been t...	1

模型：句子情感分类


“a visually stunning rumination on love”
Movie Review Sentiment Classifier
positive

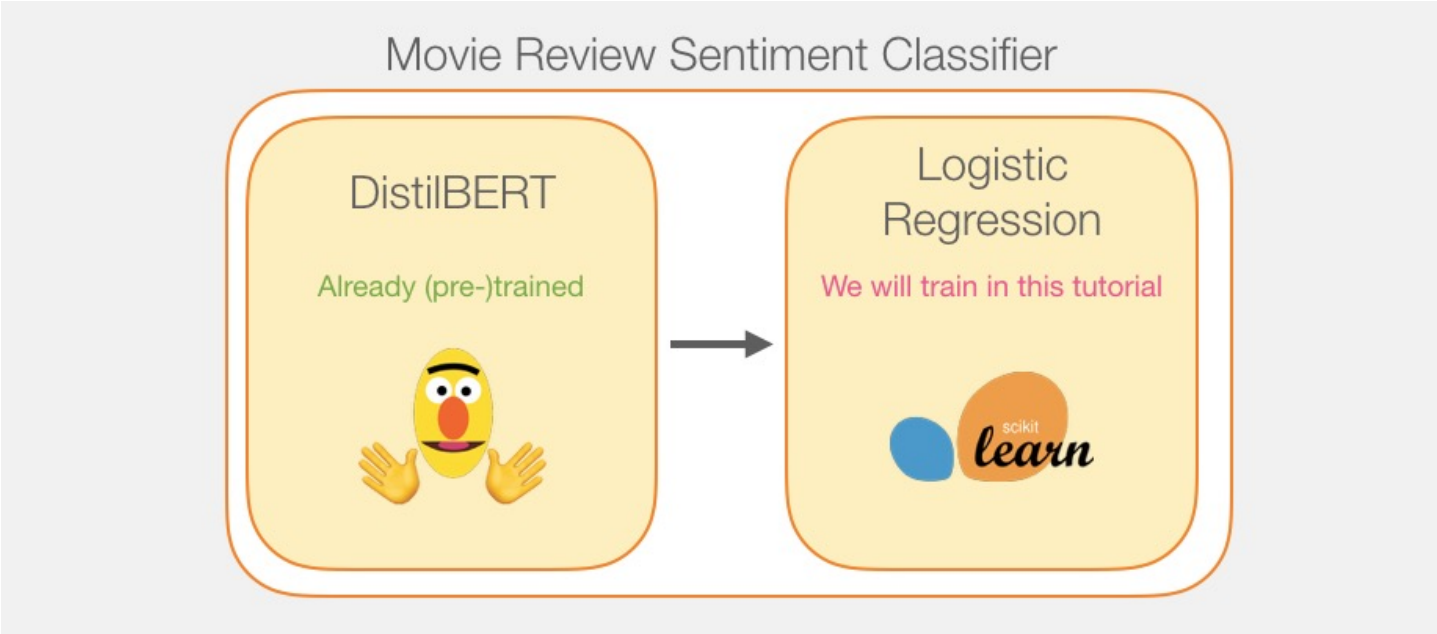
模型：句子情感分类

DistilBERT ：更小版本的 BERT 模型，保留了 BERT 能力，比 BERT 更小更快。



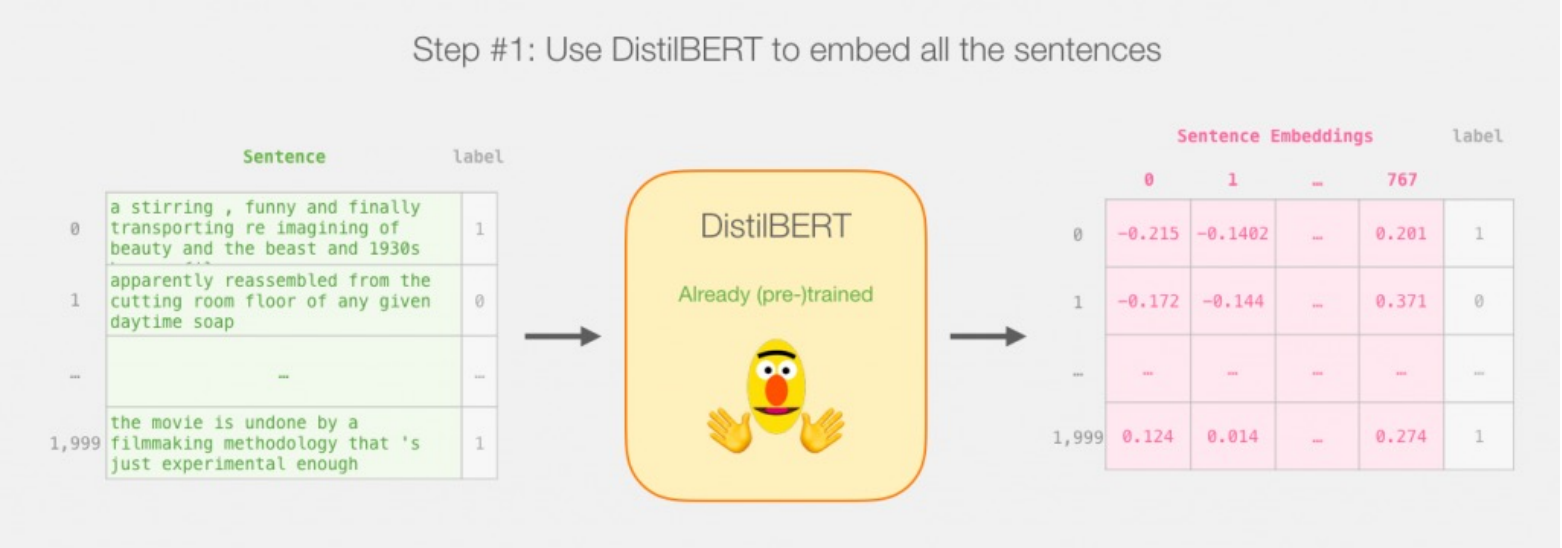
模型的训练

只训练分类器 — 逻辑回归模块



模型的运行过程

首先使用预训练的 distilBERT 模型为句子生成句向量。

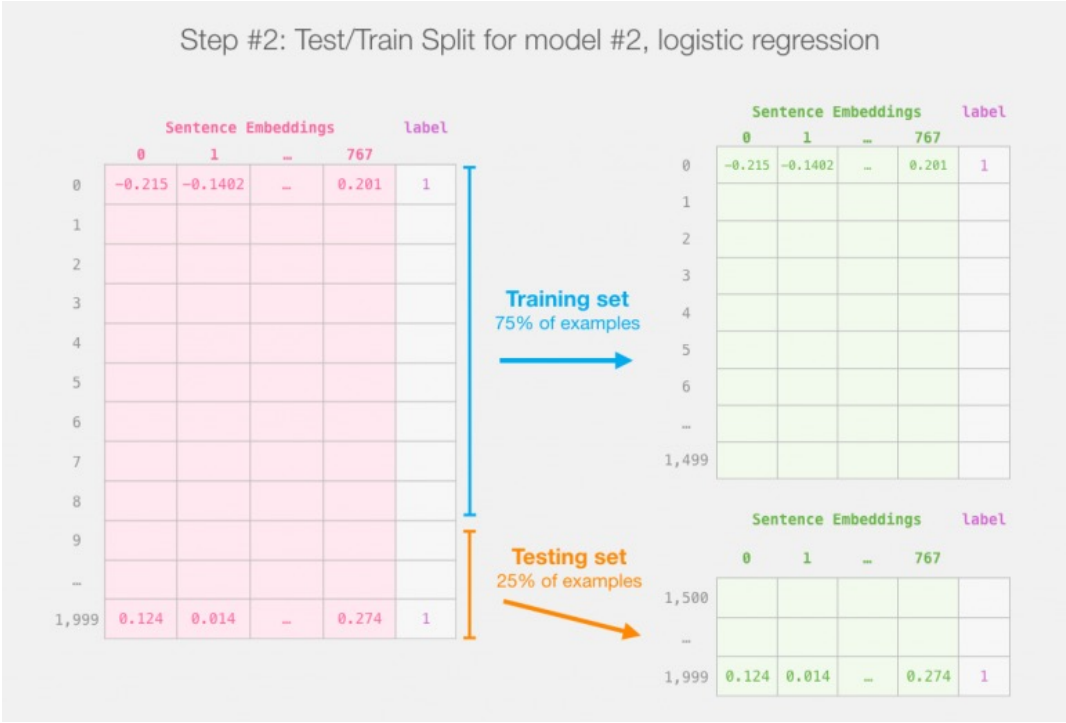


模型的运行过程

使用 Scikit Learn 工具包

将整个数据集分成

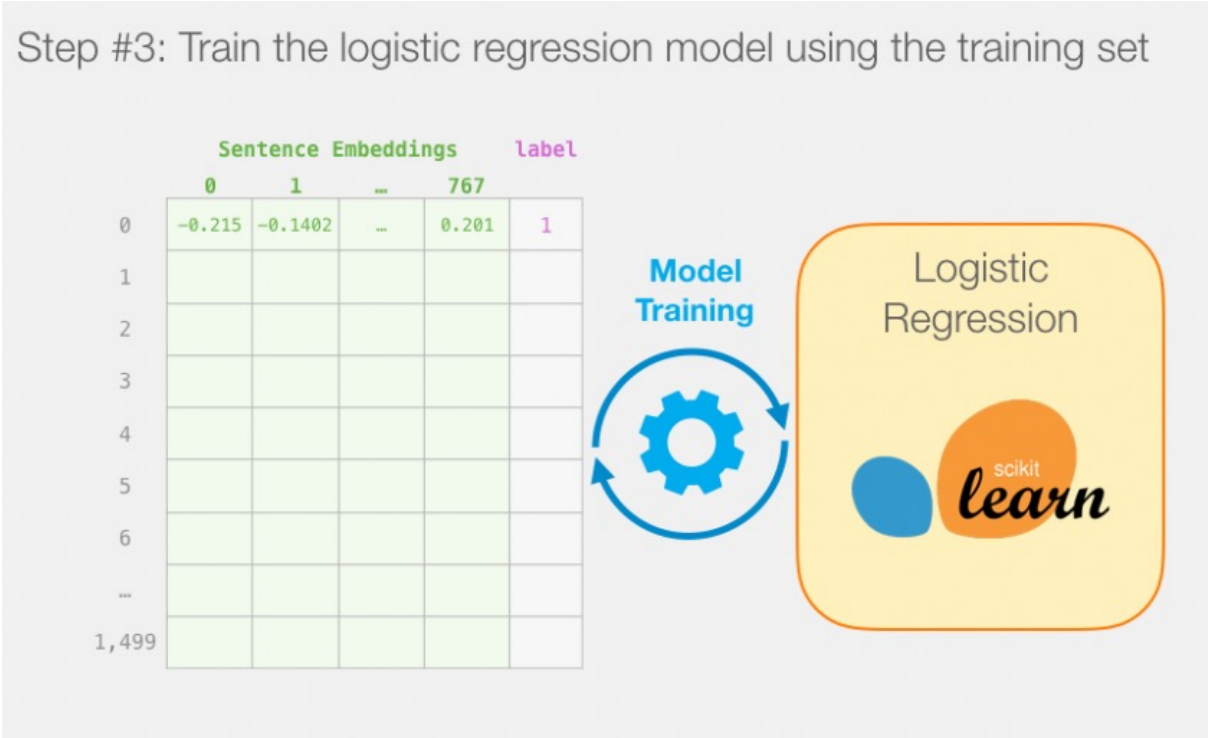
train/test 数据集



模型的运行过程

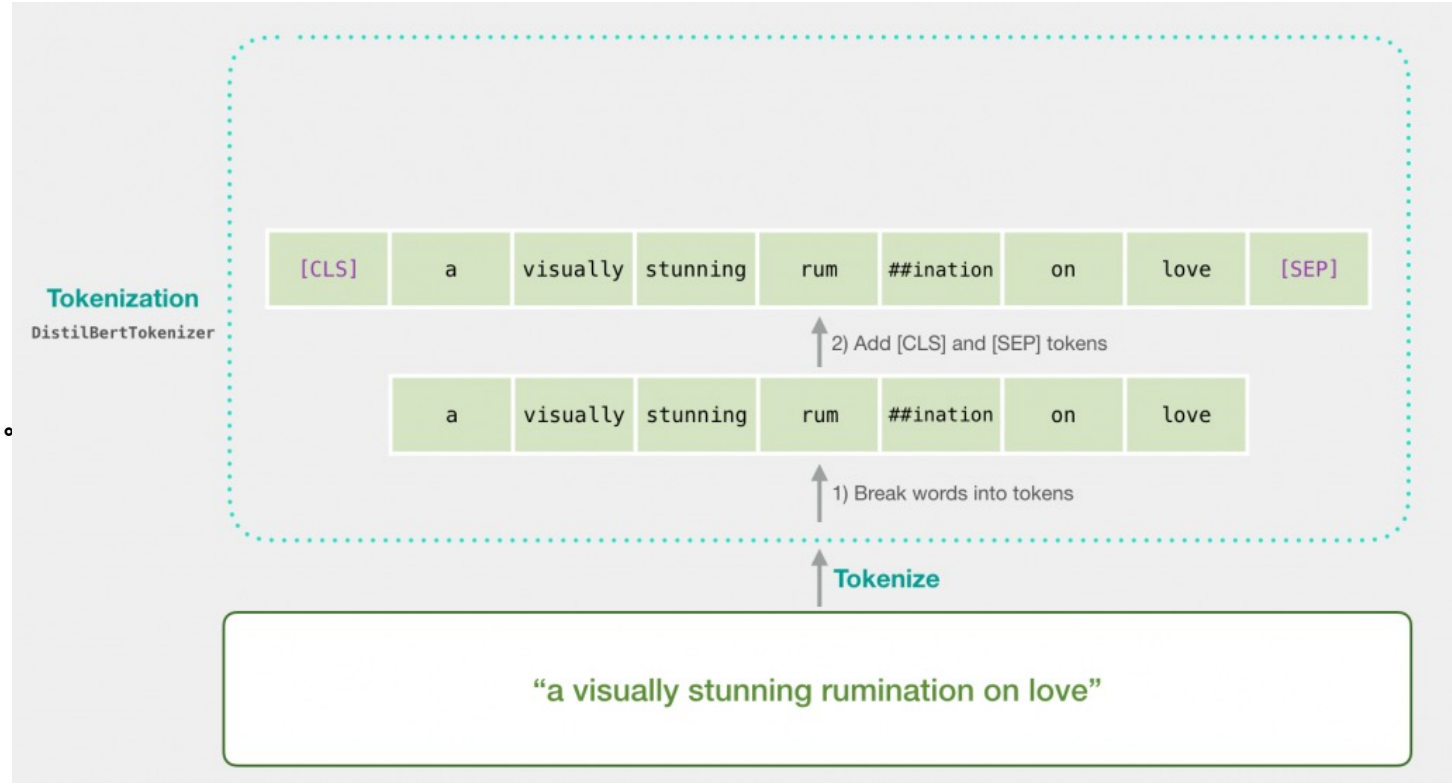
Logistic Regression

模型训练



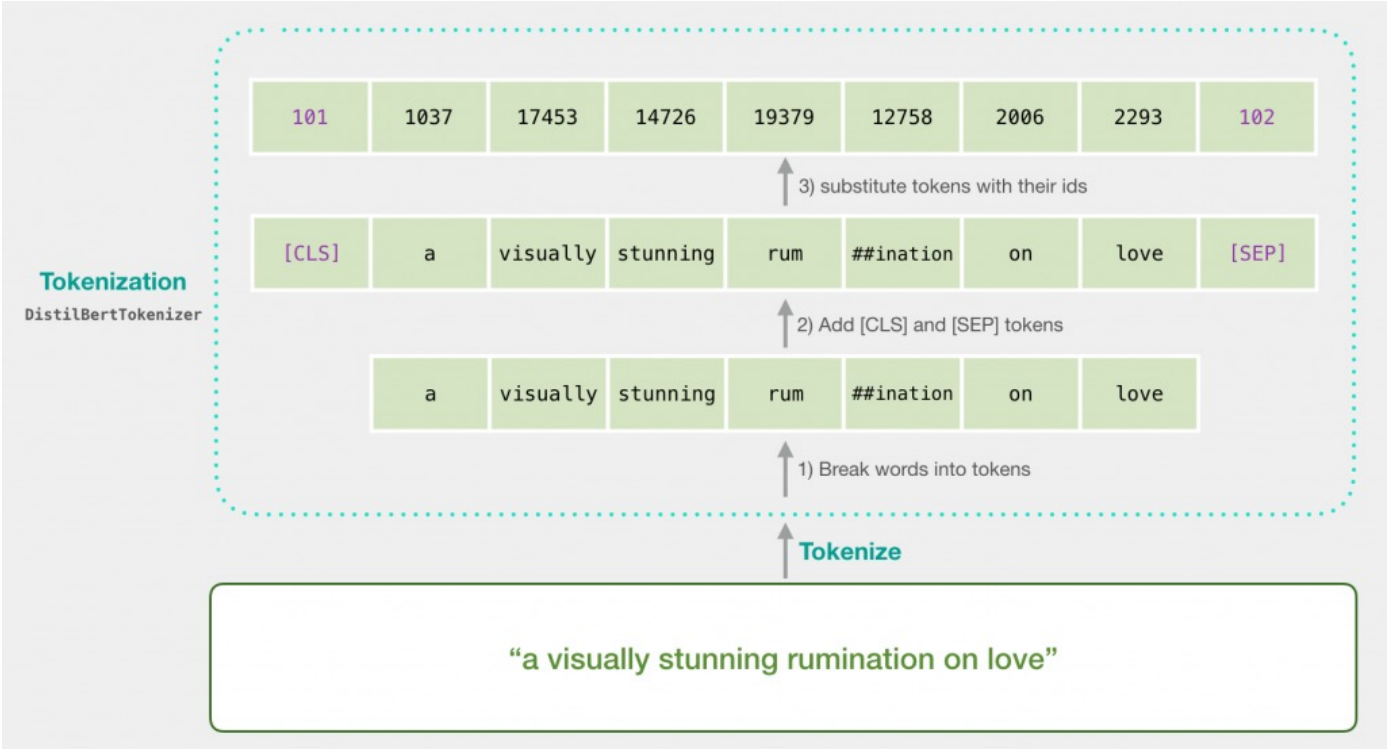
模型的数据处理过程

1. 用 BERT 的分词器（tokenizer）将句子分成 tokens；
2. 添加特殊的 tokens 用于句子分类任务（在句子开头加上 [CLS] 在句子结尾加上 [SEP]）。



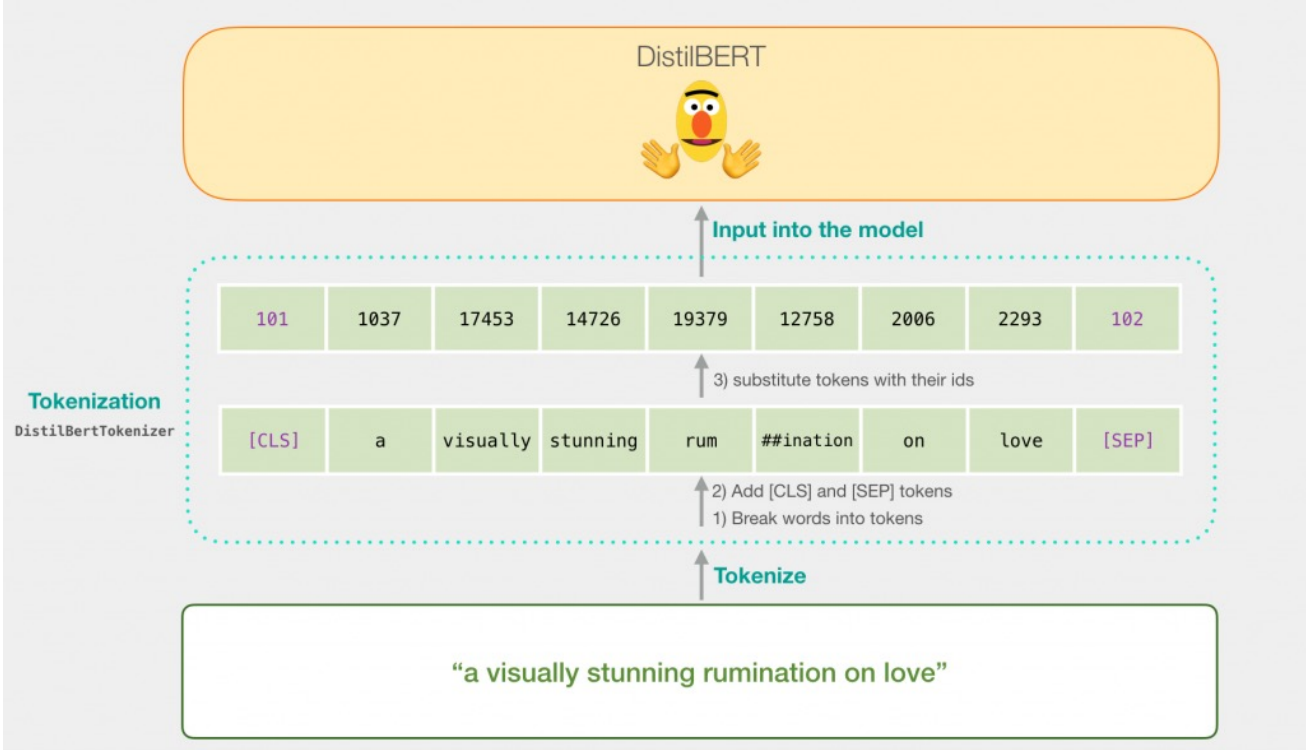
模型的数据处理过程

3. 分词器（tokenizer）将每个 token 替换成 embedding 表中的ID，embedding 表是我们预训练模型自带的。



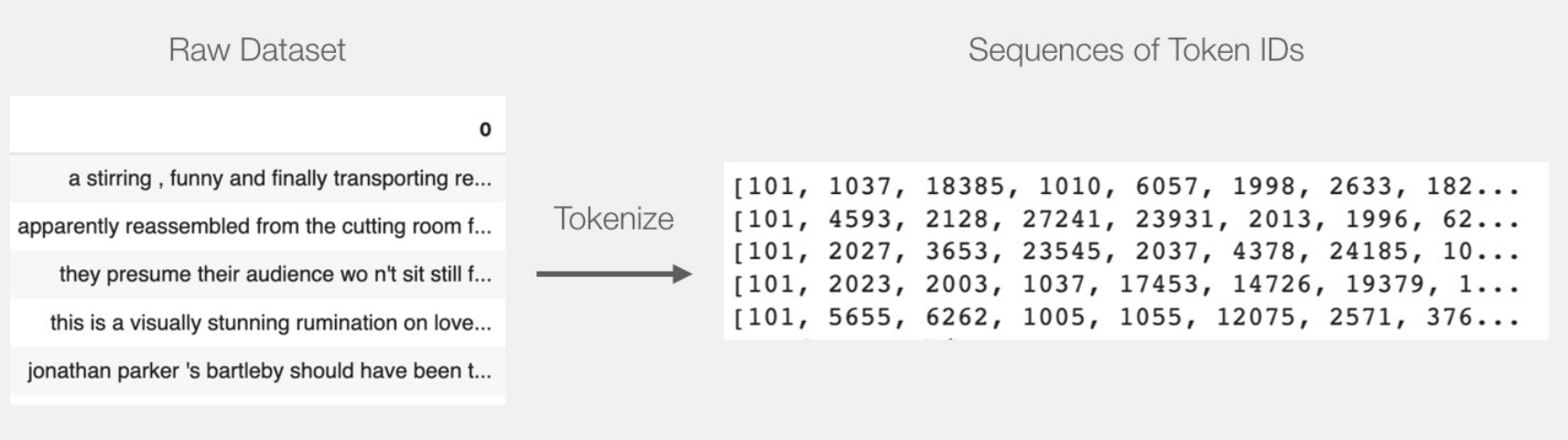
模型的数据处理过程

```
tokenizer.encode("
a visually stunning
rumination on love",
add_special_tokens=True)
```



模型的数据处理过程

Tokenization：将语句变为id序列



模型的数据处理过程

Padding：填充‘0’让语句长度一致

padded = np.array([i + [0](max_len-len(i)) for i in tokenized.values])*

```
padded[0]
```

```
array([  101,  1037, 18385,  1010,  6057,  1998,  2633, 18276,  2128,
        16603,  1997,  5053,  1998,  1996,  6841,  1998,  5687,  5469,
         3152,   102,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0])
```

模型的数据处理过程

Masking：在self-attention计算时不考虑padding的部分

attention_mask = np.where(padded != 0, 1, 0)

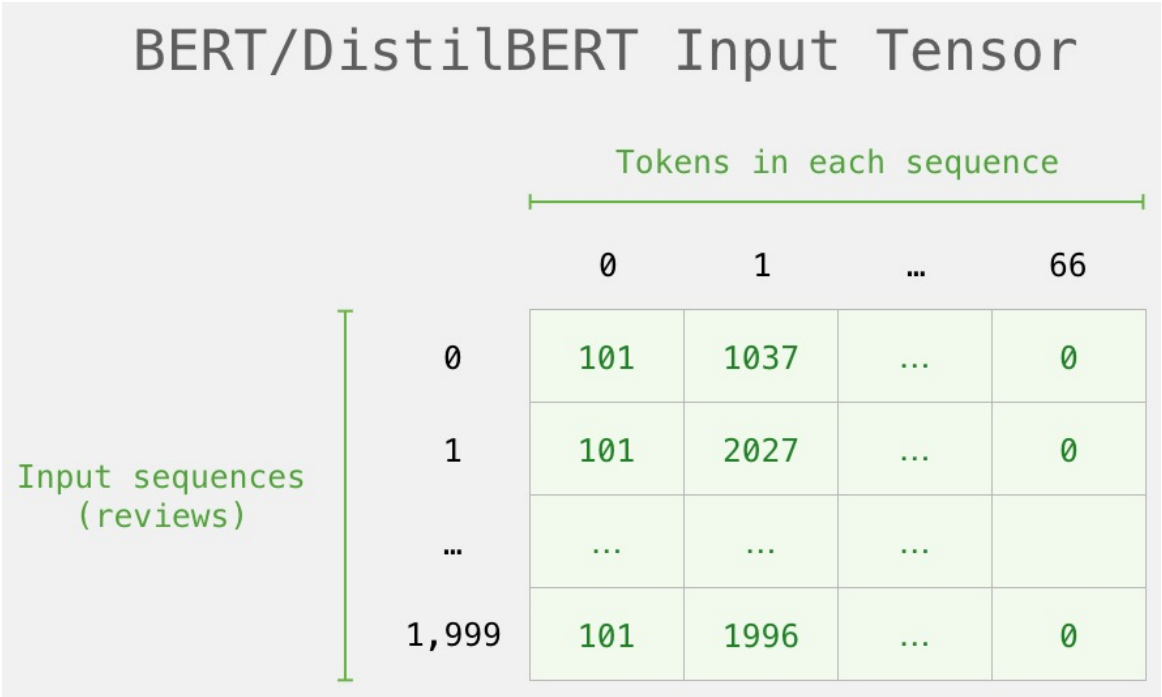
```
attention_mask[0]
```

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

模型的数据处理过程

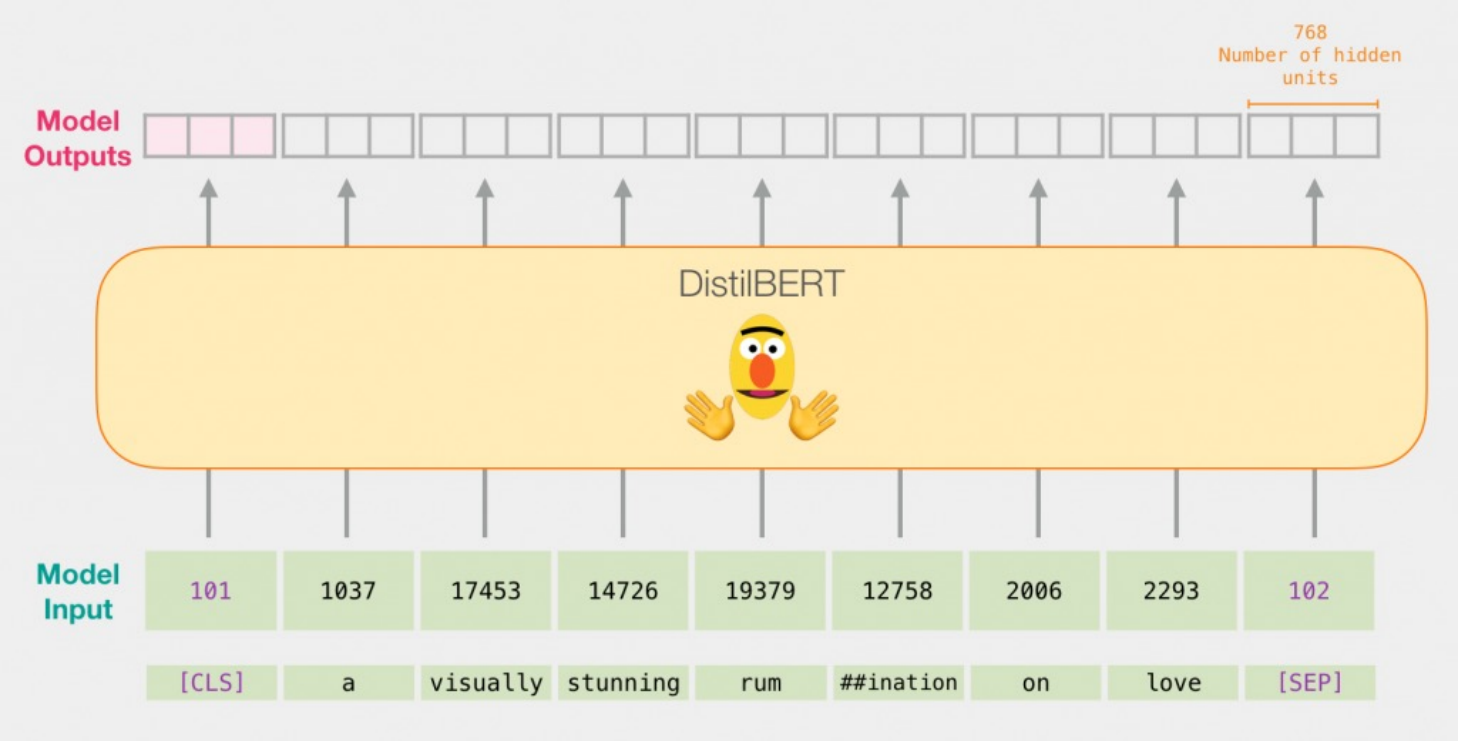
将处理好的数据传给

BERT模型

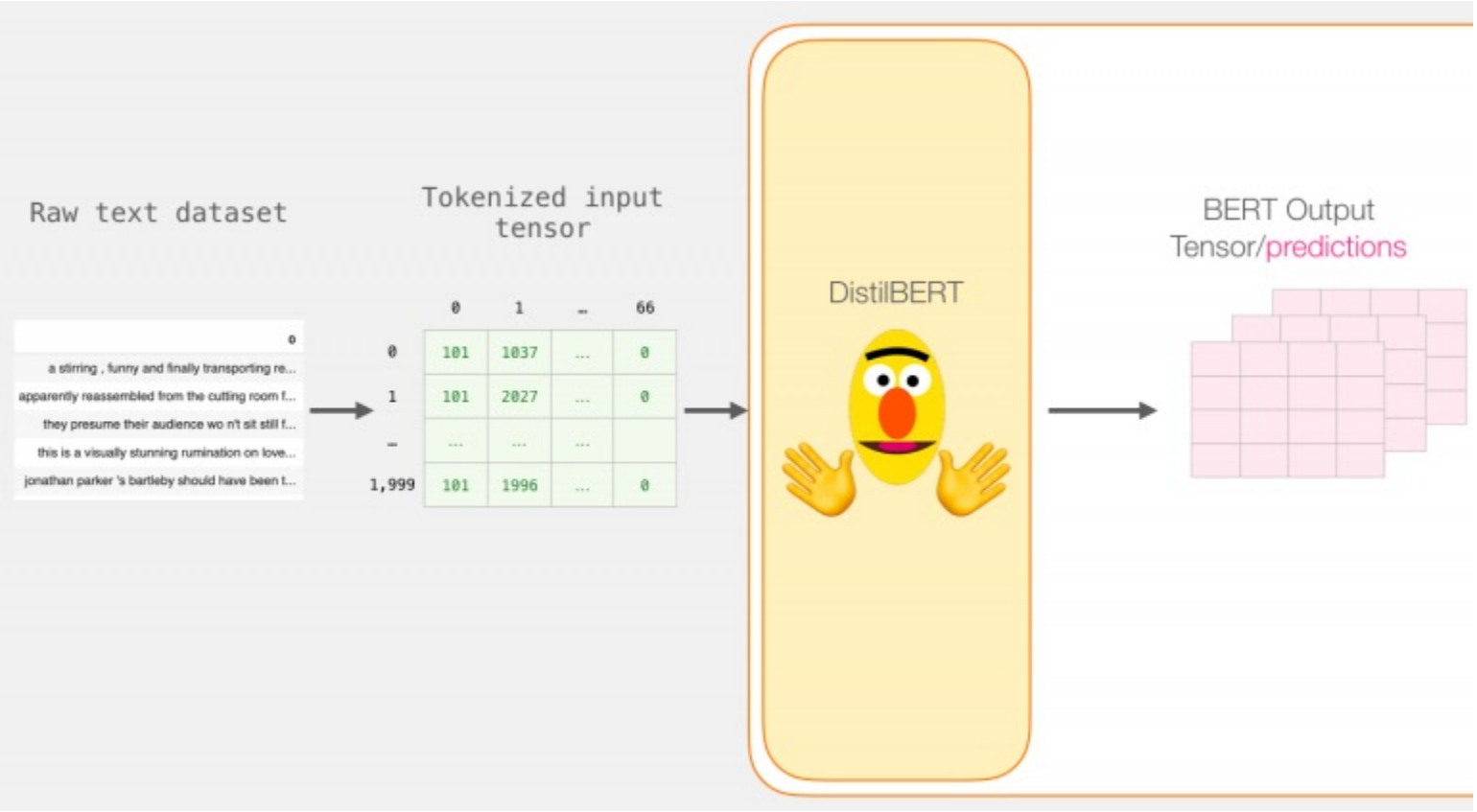


DistilBERT 中的流程

每个 token 对应的
输出是一个
768维的向量

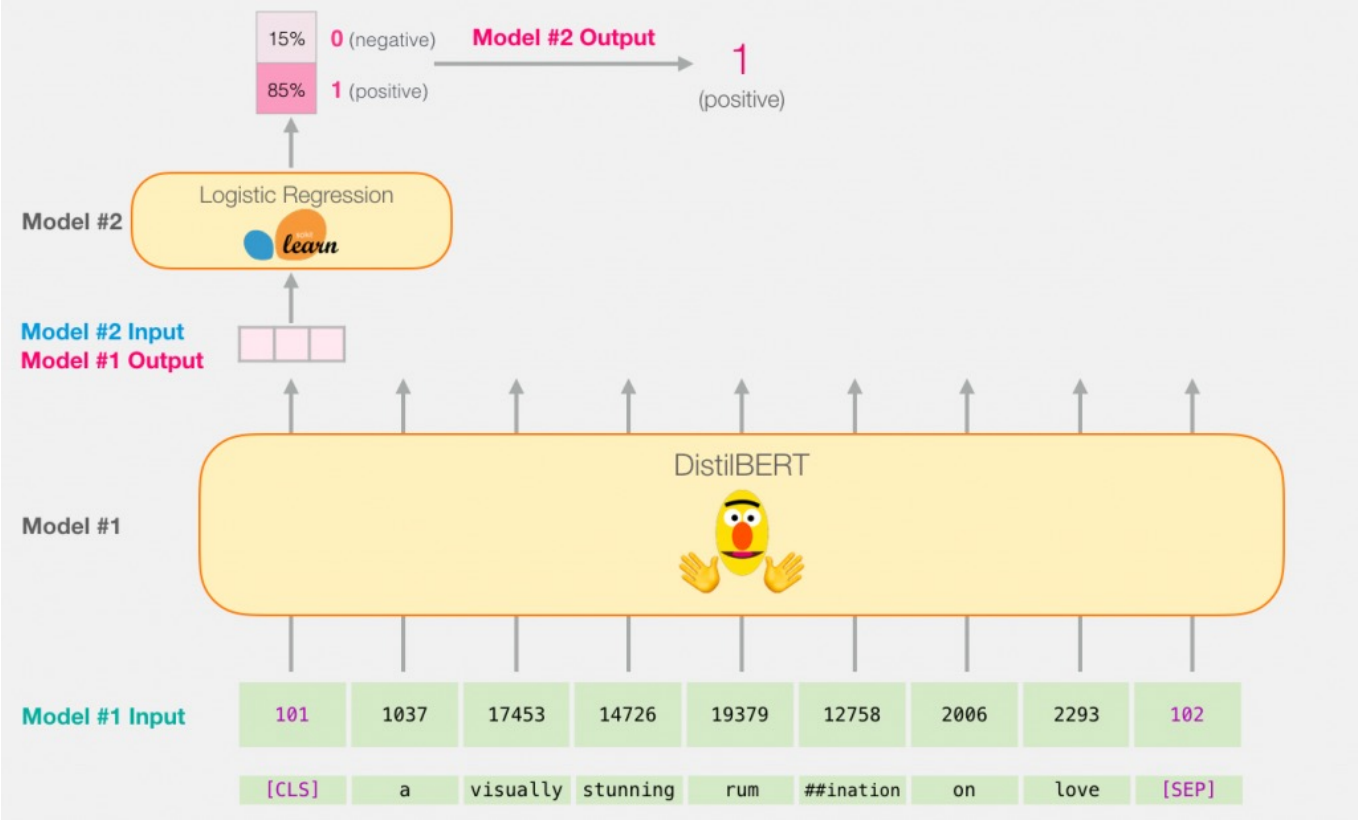


DistilBERT 中的流程



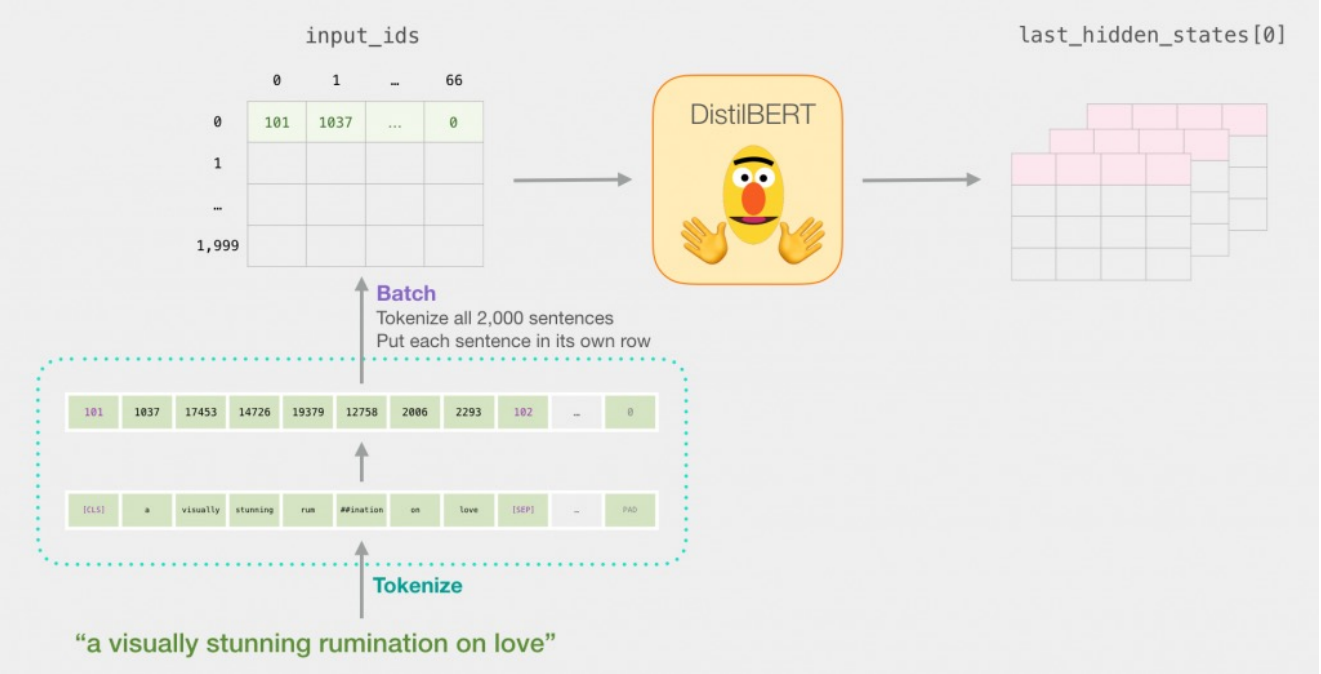
DistilBERT 中的流程

只取第一个向量（与 [CLS] 对应的向量）作为 logistic regression 的输入

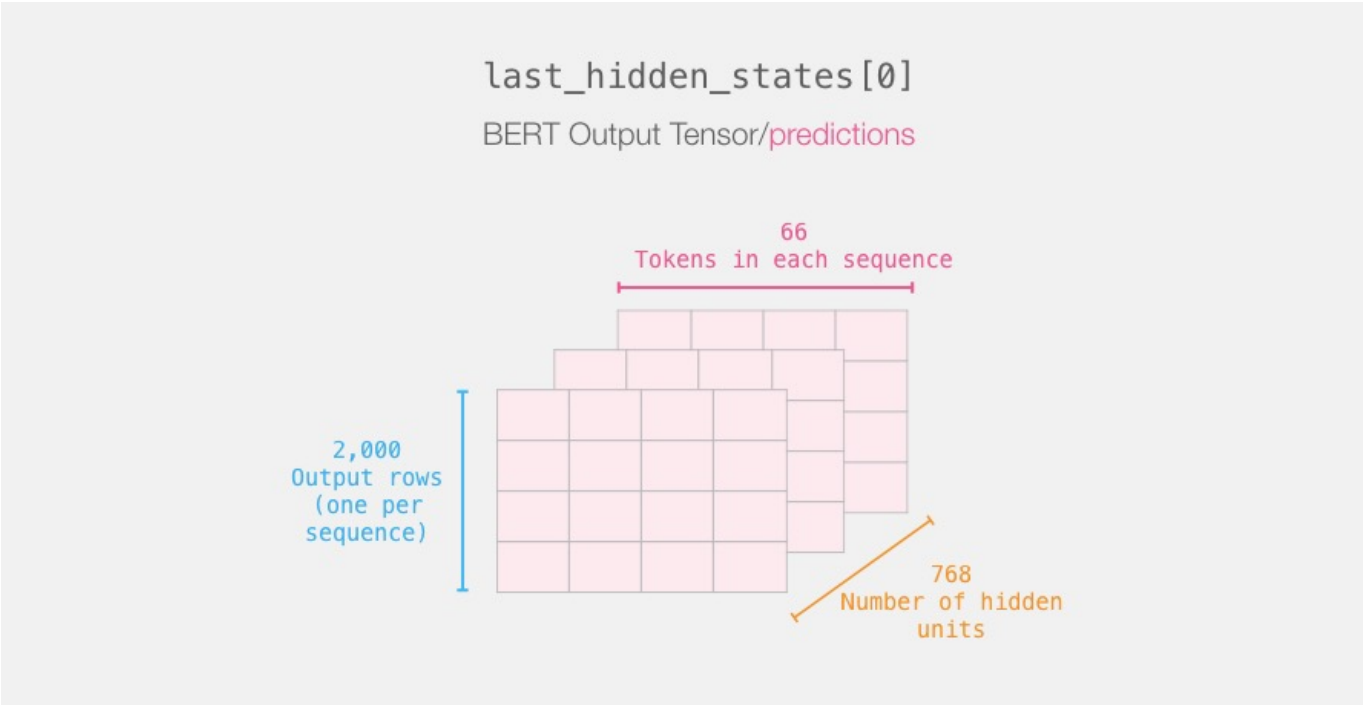


DistilBERT 中的流程

完整的流程



DistilBERT 中的流程

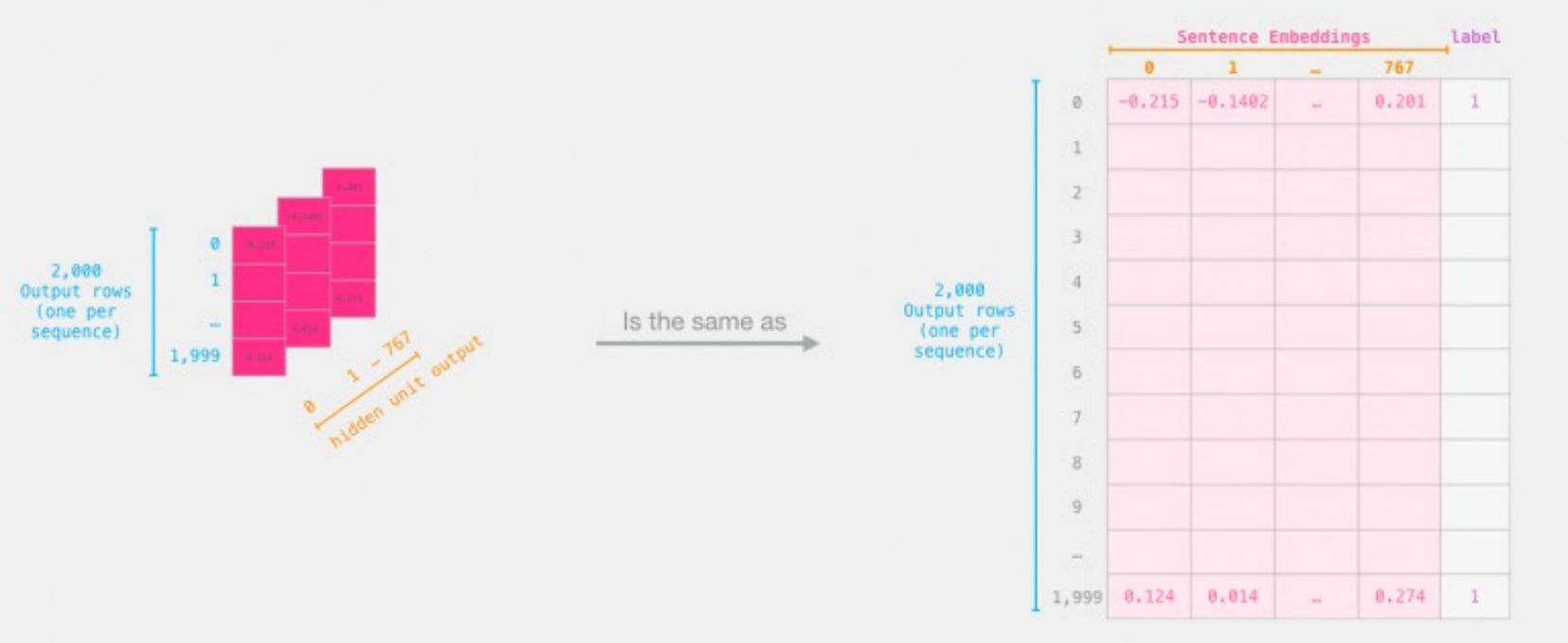


DistilBERT 中的流程

只取[CLS]对应的编码输出

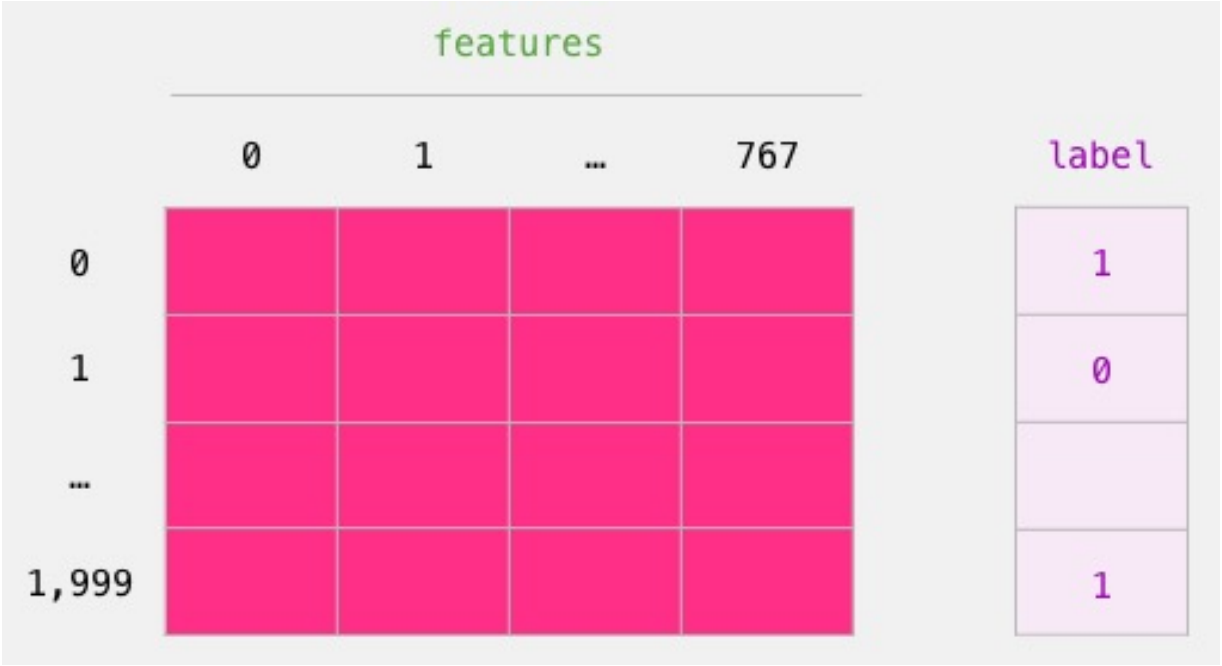
DistilBERT 中的流程

features：包含数据集中所有句子的向量的2维 numpy 数组



Logistic Regression 中的流程

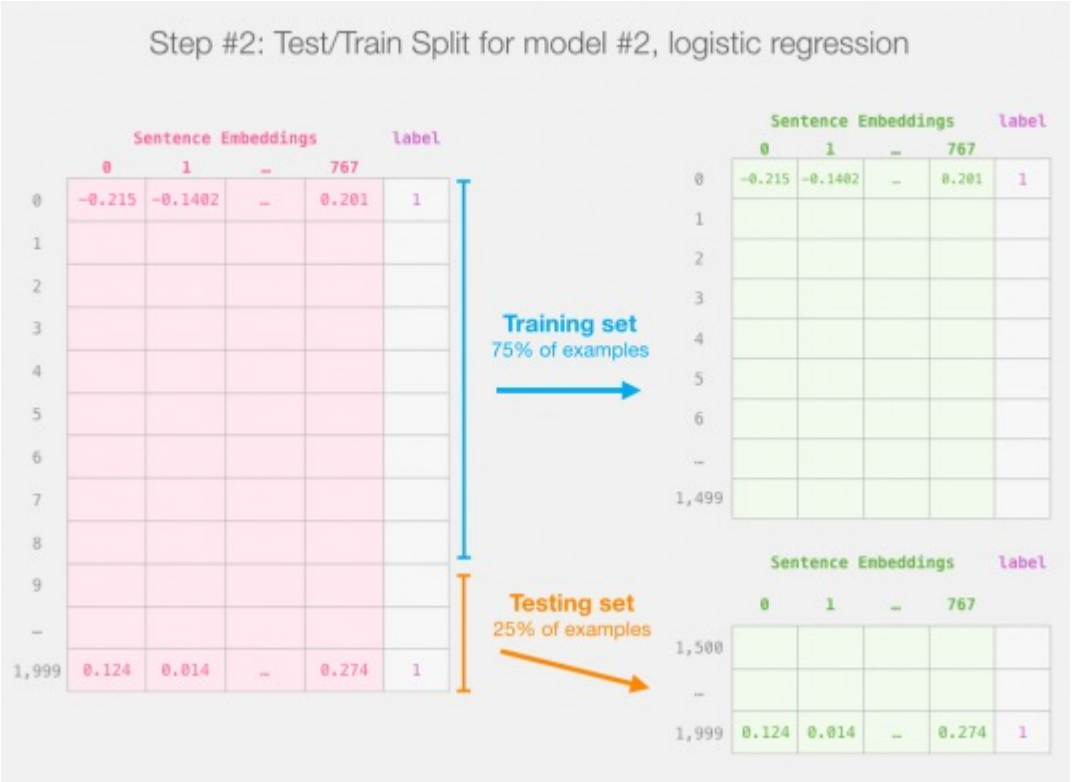
训练LR模型的数据



Logistic Regression 中的流程

训练和测试

数据的划分



Logistic Regression 中的流程

LR分类模型

训练和测试

```
lr_clf = LogisticRegression()
lr_clf.fit(train_features, train_labels)
```

```
LogisticRegression()
```

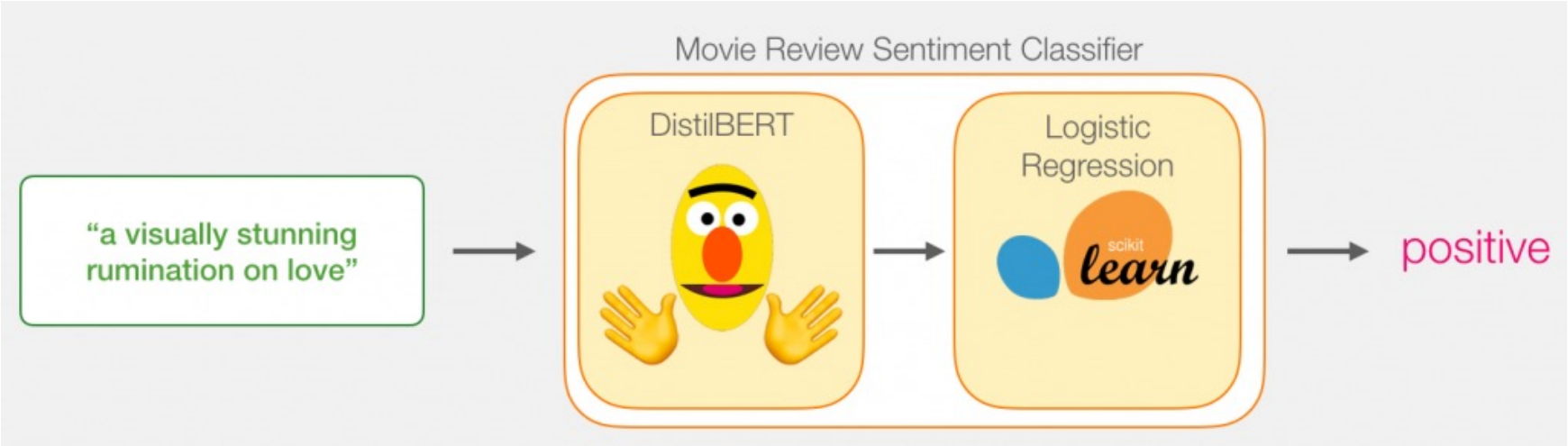
```
lr_clf.score(test_features, test_labels)
```

```
0.84
```

```
lr_clf.predict(test_features)
```

```
array([0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1,
       1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1,
       1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0,
       1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
       1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0,
       1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0,
```

整体的流程


Movie Review Sentiment Classifier
"a visually stunning rumination on love"
DistilBERT
Logistic Regression
scikit learn
positive

问题及讨论

Q&A